

PROBLEM. Countries to Visit (contributed by J. Contreras) [Score: 50/50]

Traveling fosters a medium to build human connections with one another by learning about culture, food, new places, music, and the way how people live. When you take time away from the stresses of work and daily life, it can improve your physical and mental health, motivation, relationships, job performance and perspective.

For this exam, you will input your favorite country of destination in a 2D array and define four(4) functions that will allow your program to:

1. Accept name of countries as string input and store it to a 2D array (5 pts)
2. Search a country from the elements of a 1D Array (5 pts)
3. Search a country from the elements of the 2D Array (10 pts)
4. Get the frequency of the string in the 2D Array (15 points)
5. Sort the country names in the 2D Array by row (15 points)

ENTER THE ARRAY ELEMENTS OF THE 2D ARRAY OF STRINGS

Create a function that will allow the user to enter the name of the countries to store in a 2D array as shown in the figure below: Note that samples shown are based on 3 rows and 3 columns to initialize in the 2D array.

```
Enter countries you want to visit:
Japan
Korea
Canada
USA
Sweden
Spain
Italy
Malta
Austria
```

SEARCHING OF COUNTRY FROM THE 2D ARRAY OF STRINGS

Search the name of the country from the 2D array and determines the row and column where is it found. If not found, the row and column values should be -1. Limit the functions to use from **string.h** to the 4 string functions part of the course material. Sample calls and displays from `main()` are as follows:

```
Sweden found in row 1 col 1
Thailand not found in your list of favorite countries
```

GETTING THE FREQUENCY OF EACH ELEMENT IN THE 2D ARRAY

Store the unique elements of the 2D array to the 1D array. Then, for each of countries in the 1D array, store the number of times (frequency) that country appeared in the 2D array. The frequencies should be stored into a separate 1D array of integers. Note that the country stored in index 0 of the 1D array of strings should have the corresponding frequency in index 0 of the 1D array of integers, and so on. Limit the functions to use from **string.h** to the 4 string functions part of the course material. The following is a sample result if all entries in the 2D array are unique. Refer to the output2.txt for sample results given 2D array has duplicate entries (from input2.txt).

```
Frequency of each string (country name) from the 2D array:
Japan : 1
Korea : 1
Canada : 1
USA : 1
Sweden : 1
Spain : 1
Italy : 1
Malta : 1
France : 1
```

ROW-WISE Sorting

Sorting is arranging of letters /numbers in an ascending order or descending order. It is a method of organizing data that is especially useful for programming computers. For this exam, the target result of the function is to have each row sorted in increasing order (based on ASCII value).

```
Sorted Array of String in Row-Wise technique
Canada    Japan    Korea
Spain     Sweden   USA
France    Italy    Malta
```

To answer this programming problem, you are given the following accompanying files:

- **countries.h** - the header file that contains the macro definitions and function prototypes that you need to define in LASTNAME-Countries.c

- **LASTNAME-Countries.c** - the skeleton file where you will encode your functions
- **main.c** - the C program that contains the main() that you need to run to test your functions
- **input1.txt** - a text file containing a sample input with unique countries
- **input2.txt** – second text file containing a sample input for countries with countries entered more than once
- **output1.txt** - a text file containing the expected output of the program given the input2.txt
- **output2.txt** –second text file containing the expected output of the program given the input2.txt

RUN YOUR PROGRAM WITH INPUT and OUTPUT REDIRECTION:

Run your exe file in the command line with input redirection. You can store the result of your program into a text file by output redirection, For example, if after **compiling main.c**, you have the executable prog.exe, then you can run your executable program with output redirection as:

```
C:\CCPROG2> prog < input2.txt > SANTOS-ACTUAL.txt
```

The output of the exe file will be stored in SANTOS-ACTUAL.txt file. A correct program using the input2.txt for input redirection should produce the same set of values that are in output2.txt provided.

TESTING & SCORING:

- Your program will be compiled via gcc -Wall, using C99 standard. Thus, for each function that does not compile successfully, the score for that function is 0.
- Your program will be tested by your instructor with other TEXT FILES (contents are different from the ones given to you) and with function calls of different parameter values.
- Full credit will be given for the function only if the student's implementation are all correct for all the test values used by the instructor during checking AND only if the student's implementation complied with the requirement and did not violate restrictions. **Brute force solution will not be given any credit.** Deductions will be given if not all test cases produce correct results OR if restrictions were not followed.

DELIVERABLES:

Submit/upload two files via Canvas before the indicated deadline:

1. your LASTNAME-Countries.c source file
2. your LASTNAME-Actual.txt output text file

Don't forget to change the filenames to your own last name